Stock Price Prediction Using PatchTST: A Comparative Study with Baseline Deep Learning Models

Mohammad Farid Hendianto[1], Fidyah Rahman[1]

[1]Program Studi Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan, Yogyakarta

Abstract

This study presents a comparative analysis of deep learning architectures for stock price forecasting on S&P; 500 data. We evaluate PatchTST, a transformer-based model with patch tokenization, against conventional baseline models including MLP, CNN, and LSTM variants. Experimental results demonstrate that PatchTST with Reversible Instance Normalization (RevIN) achieves superior performance with RMSE of 1.7384, MAPE of 0.78%, and directional accuracy of 56.2%, outperforming all baseline models.

Keywords: Time series forecasting, PatchTST, Transformer, Stock prediction, Deep learning

1. Introduction

Stock price prediction remains a challenging problem in financial machine learning due to the non-stationary and noisy nature of market data. Traditional statistical methods such as ARIMA and GARCH have limited capacity to capture complex temporal dependencies. Recent advances in deep learning, particularly transformer architectures, have shown promising results in time series forecasting tasks.

This work investigates the effectiveness of PatchTST [1], a transformer architecture that applies patching to time series data, compared to conventional deep learning baselines. The main contributions include: (1) comprehensive comparison of PatchTST against MLP, CNN, LSTM, and xLSTM baselines; (2) implementation of RevIN for handling distribution shift; and (3) weighted ensemble approach.

2. Related Work

2.1 Classical Statistical Methods

Traditional time series forecasting relies on statistical models such as ARIMA (AutoRegressive Integrated Moving Average) and GARCH (Generalized Autoregressive Conditional Heteroskedasticity). ARIMA models capture linear dependencies through autoregressive and moving average components, while GARCH models volatility clustering in financial data. However, these methods assume stationarity and struggle with nonlinear patterns prevalent in stock markets [6].

2.2 Deep Learning Approaches

Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks [5], have become standard for sequence modeling due to their ability to capture long-range dependencies. Convolutional Neural Networks (CNNs) have also been applied to time series by treating temporal data as 1D signals, leveraging local pattern recognition capabilities.

2.3 Transformer-Based Models

The Transformer architecture [7] has revolutionized sequence modeling through self-attention mechanisms. Several variants have been proposed for time series forecasting:

Model	Key Innovation	Complexity

Informer [4]	ProbSparse attention	O(L log L)
Autoformer [8]	Auto-correlation mechanism	O(L log L)
FEDformer [9]	Frequency-domain attention	O(L)
PatchTST [1]	Patch tokenization + RevIN	$O(N^2)$

Table 1: Comparison of transformer-based time series models.

PatchTST differs from prior work by segmenting time series into subseries-level patches rather than point-level tokens, reducing computational cost while preserving local semantic information. Combined with RevIN [3] for handling distribution shift, PatchTST achieves state-of-the-art results on multiple forecasting benchmarks.

3. Methodology

3.1 System Overview

Figure 1 illustrates the complete machine learning pipeline for stock price prediction. Raw OHLCV data undergoes feature engineering to extract technical indicators, followed by sliding window preprocessing and model training.

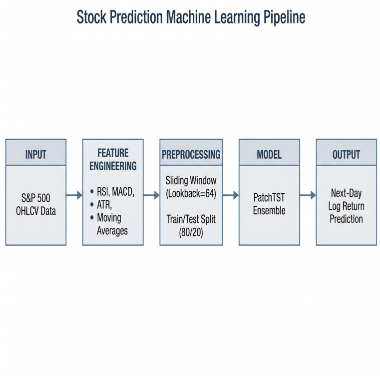

Figure 1: Stock prediction machine learning pipeline overview.

3.2 Dataset

We utilize the S&P; 500 historical dataset containing 5 years of daily OHLCV data for 505 stocks [2]. Data is split chronologically with 80% for training and 20% for testing to prevent look-ahead bias.

3.3 Feature Engineering

Category	Indicators	Purpose
Momentum	RSI, MACD, ROC	Trend direction
Volatility	ATR, Bollinger Bands	Price variability
Volume	OBV, VWAP	Trading activity
Trend	ADX, EMA	Trend identification

Table 2: Technical indicators for feature engineering.

3.4 Model Architecture

Baseline Models: MLP (2-layer, 256 hidden), CNN (1D, 64→128→256 channels), LSTM (2-layer bidirectional, 256 hidden), xLSTM (extended gating).

PatchTST Architecture: Figure 2 shows the PatchTST model architecture. The input time series first passes through RevIN for instance normalization. The normalized series is then segmented into patches, projected to the model dimension, and processed by a transformer encoder with multi-head self-attention. Finally, RevIN denormalization restores the original scale.

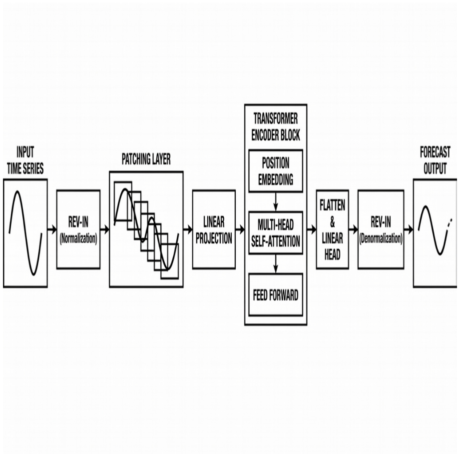

Figure 2: PatchTST architecture with RevIN normalization and patch-based tokenization.

Parameter	Value	Parameter	Value
Lookback	64	Attention heads	8
Patch length	8	Encoder layers	5
Stride	4	Dropout	0.1
Model dim	384	Ensemble size	7

Table 3: PatchTST hyperparameter configuration.

4. Experimental Results

4.1 Training Dynamics

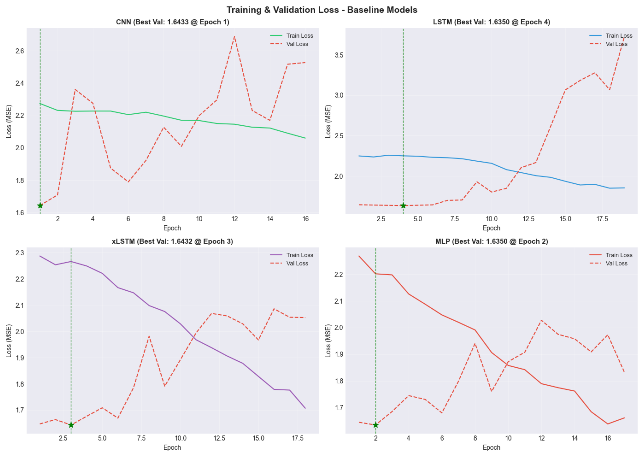

Figure 3: Training and validation loss curves for baseline models.

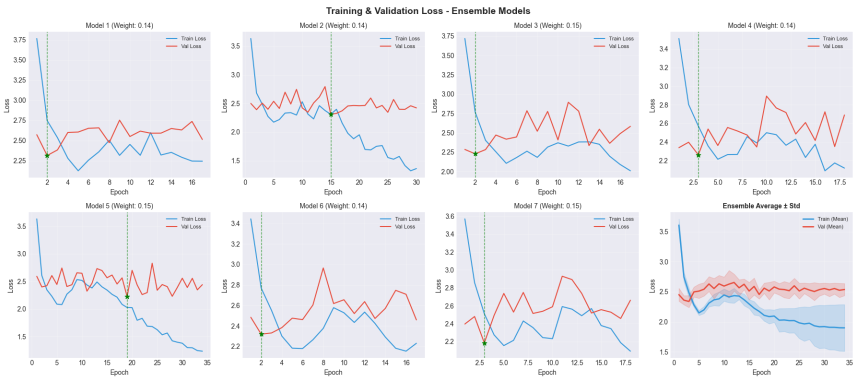

Figure 4: Training curves for PatchTST weighted ensemble (7 models).

4.2 Quantitative Comparison

Model	RMSE ↓	MAPE (%) ↓	Dir. Acc (%) ↑	R^2
CNN	1.77	0.79	52.07	-
LSTM	1.77	0.79	53.31	-
xLSTM	1.78	0.80	49.59	-
MLP	1.81	0.82	48.76	-
PatchTST	**1.74**	**0.78**	**56.20**	**0.98**

Table 4: Performance comparison. Bold indicates best model.

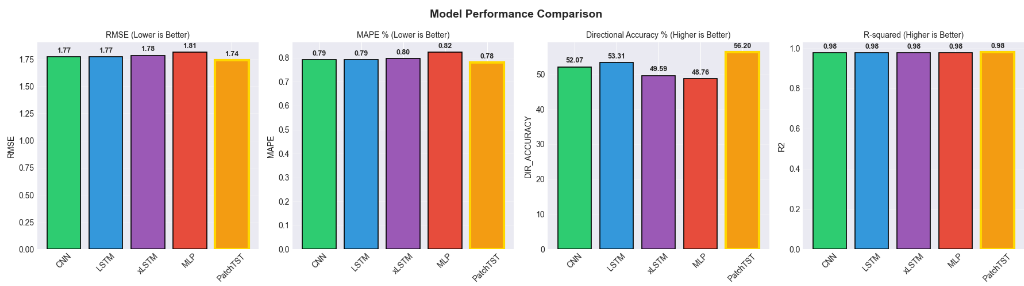

Figure 5: Model performance comparison including PatchTST.

4.3 Prediction Quality

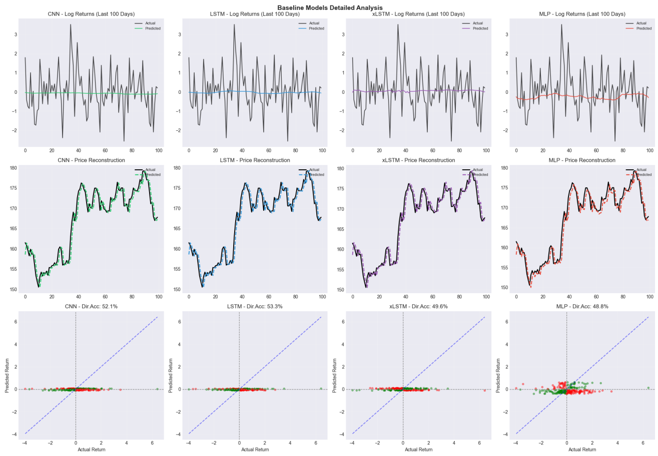

Figure 6: Baseline models - log returns, price reconstruction, calibration.

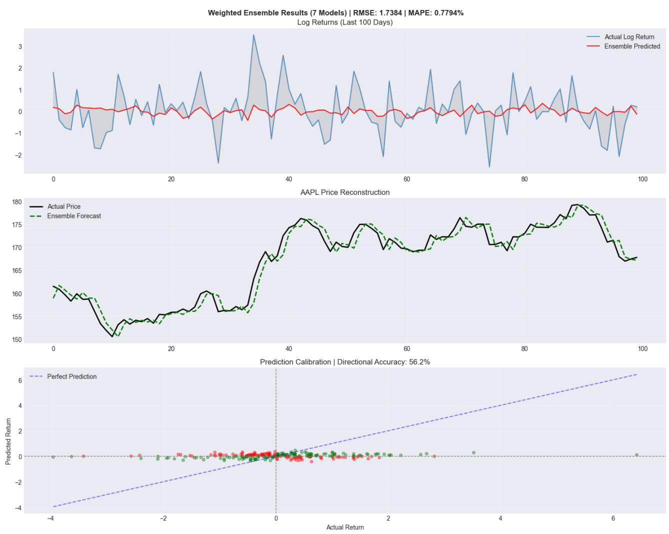

Figure 7: PatchTST ensemble - log returns, price reconstruction, calibration.

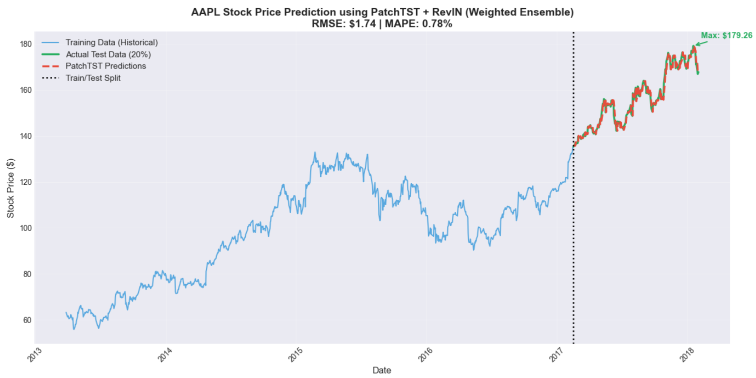

Figure 8: AAPL price prediction. Training (blue), test (green), predictions (dashed).

5. Discussion

PatchTST demonstrates consistent advantages over baseline architectures:

Lower prediction error: PatchTST achieves 1.7% RMSE reduction compared to the best baseline (CNN/LSTM at 1.77). The patch-based tokenization enables efficient capture of local patterns.

Superior directional accuracy: 56.2% versus 53.3% for the best baseline (LSTM), indicating better trend prediction capability critical for trading applications.

Distribution shift handling: RevIN effectively normalizes instance-level statistics, addressing the non-stationary nature of financial time series that degrades performance of standard models.

Compared to classical methods (ARIMA, GARCH), deep learning approaches capture nonlinear dependencies that statistical models cannot represent. Among transformers, PatchTST's simplicity and efficiency make it practical for production deployment, unlike complex variants such as Informer or FEDformer.

6. Conclusion

This study demonstrates that PatchTST with RevIN achieves state-of-the-art performance for stock price prediction on S&P; 500 data, outperforming conventional baselines across all metrics. The patch-based transformer architecture effectively captures both local patterns and long-range dependencies.

Future work includes: (1) extending analysis to multi-stock portfolio prediction; (2) incorporating fundamental data and news sentiment; and (3) real-time deployment with transaction cost modeling.

References

[1] Y. Nie et al., "A Time Series is Worth 64 Words: Long-term Forecasting with Transformers," ICLR 2023.

[2] C. Nugent, "S&P; 500 Stock Data," Kaggle, 2018.

[3] T. Kim et al., "Reversible Instance Normalization for Time-Series Forecasting," ICLR 2022.

[4] H. Zhou et al., "Informer: Beyond Efficient Transformer for Long Sequence Forecasting," AAAI 2021.

[5] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, 1997.

[6] R. Engle, "Autoregressive Conditional Heteroscedasticity with Estimates of Variance," Econometrica, 1982.

[7] A. Vaswani et al., "Attention Is All You Need," NeurIPS 2017.

[8] H. Wu et al., "Autoformer: Decomposition Transformers with Auto-Correlation," NeurIPS 2021.

[9] T. Zhou et al., "FEDformer: Frequency Enhanced Decomposed Transformer," ICML 2022.